



06

Uso this

Transcripción

[00:00] Ahora, quizás en algunos tutoriales o documentación, tú has visto que delante de los métodos tenemos la palabra reservada `public`. Esto es un modificador de acceso, al igual que ya quizás te has dado cuenta de está presente aquí también `public static void`, `public class`. Pero este no es el momento para discutir de eso.

[00:26] Vamos a dejarlo así con `public`, pero no tiene nada que ver con nuestro comportamiento del método que estamos implementando, es totalmente independiente y tiene otras finalidades que vamos a explicar en el curso. Ahora, el código compila, no hay ningún problema aquí, pero hay algunas prácticas en las que queremos hacer un poco de énfasis, buenas prácticas mejor dicho.

[00:53] Y es que por ejemplo, si yo hubiera llamado aquí esta variable de saldo y yo reemplazo aquí, por ejemplo, vemos pues que `public void depositar` sería igual `double saldo + saldo`, y aquí también es `saldo` porque `saldo` es el nombre de mi variable pero también es el nombre de mi parámetro y comienza a quedar todo muy confuso.

[01:23] Esto lo vamos a ver mucho más cuando toquemos ya el tema de constructores, que es otro concepto que vamos a ver ya de aquí a muy poco, pero en este caso nosotros vamos a usar otra palabra más reservada de Java. Para hacerlo un poco más dinámico, vamos a hacer una pequeña analogía.

[01:49] Yo estoy aquí con mi clase cuenta. Esta es mi cuenta, perfecto, y cuando yo estoy depositando, estoy depositando a mi cuenta, por lo tanto en el método depositar, él deposita a esta cuenta. Entonces, vamos aquí a separar, y saldo es un atributo de esta cuenta. ¿Cierto? Esta cuenta. Vamos a separar un poco así para que quede más práctico. Listo.

[02:30] Entonces saldo hace referencia al saldo de esta cuenta de aquí, y esta palabra saldo de aquí también hace referencia al atributo saldo de esta cuenta. Entonces sería el saldo de esta cuenta, el saldo de esta cuenta. Y esta cuenta, como es el mismo objeto, ustedes ya saben que casi todos los lenguajes de programación están escritos en inglés, si yo lo traduzco al inglés sería como decir `this account`.

[03:14] `This account`. Perfecto. Pero sabiendo que estamos hablando ya de objetos, un lenguaje orientado a objetos, sería como especificar `this object`. ¿Cierto? Vamos retomando un poco cómo es la referencia. Sabemos que ese saldo es el saldo de esta cuenta traducido al inglés sería `this account`, y como estamos digamos en un lenguaje orientado a objetos y vamos a llamarlo de una forma genérica, cuenta es un objeto, cuenta es una clase, entonces vamos a llamarlo `this object`.

[04:00] Y aquella palabra clave reservada de Java para hacer referencia a este objeto, adivina cuál es: el conocido `this`. `This` hace referencia a este objeto. Tú estás trabajando con este objeto de aquí, entonces aquí sería `this.saldo`. Perfecto. ¿Entonces qué le estás diciendo? Vamos a ordenar un poco el código otra vez. ¿Qué le estás diciendo a Java en este momento?

[04:34] Le estás diciendo al saldo de esta cuenta, de este objeto, le vas a aumentar su saldo que tiene, o sea el saldo que ya existe, más el saldo que llega como parámetro que es este que está aquí suelto. Este saldo no es `this.saldo`. ¿Por qué? Porque simplemente él está llegando como parámetro de afuera.

[05:03] Por ejemplo si yo tuviera igual el valor antiguo valor, aquí también tengo valor, yo no podría hacer por ejemplo `this.valor`. ¿Por qué? Porque valor no existe dentro del objeto, valor no es un campo del objeto cuenta, por lo tanto yo no puedo usar el `this` aquí en este caso. Cerremos aquí, eliminamos este espacio, guardamos, ejecutamos.

[05:40] Vamos a nuestra consola, le damos play y el comportamiento sigue siendo el mismo. El uso de `this` es una práctica muy utilizada entre programadores Java. Sirve mucho para especificar a qué atributo tú estás haciendo referencia. Van a haber casos, van a haber situaciones en las cuales el nombre de tu variable sea igual al nombre de alguno de los campos, entonces el uso de `this` hace que tu código sea mucho más fácil de leer e interpretar.